Министерство образования и науки РФ

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Лабораторная работа № 1

“**Явные методы Эйлера**”

по дисциплине «Численное моделирование динамических систем, описываемых обыкновенными дифференциальными уравнениями»

**Факультет:** ПМИ

**Группа:**  ПМ-92

**Студенты:**  Иванов В., Кутузов И.

**Преподаватель:**  Вагин Д. В.

Новосибирск

2021

**Условие**

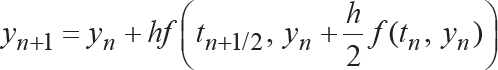
На трех сетках h = [0.1, 0.05, 0.025] решить задачу

y'=4ty

t=[0,1]

y(0)=1

с помощью трех явных схем Эйлера.



**Шаг 0.1**

|  | | **Метод 1** | | **Метод 2** | | **Метод 3** | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **t** | **y\_аналит** | **y\_числен** | **погрешн** | **y\_числен** | **погрешн** | **y\_числен** | **погрешн** |
| 0.0 | 1.0 | 1.0 | 0.00e+00 | 1.0 | 0.00e+00 | 1.0 | 0.00e+00 |
| 0.1 | 1.020201 | 1.0 | 2.02e-02 | 1.02 | 2.01e-04 | 1.02 | 2.01e-04 |
| 0.2 | 1.083287 | 1.04 | 4.33e-02 | 1.082832 | 4.55e-04 | 1.082424 | 8.63e-04 |
| 0.3 | 1.197217 | 1.1232 | 7.40e-02 | 1.196313 | 9.05e-04 | 1.194996 | 2.22e-03 |
| 0.4 | 1.377128 | 1.257984 | 1.19e-01 | 1.375281 | 1.85e-03 | 1.372334 | 4.79e-03 |
| 0.5 | 1.648721 | 1.459261 | 1.89e-01 | 1.644836 | 3.88e-03 | 1.639115 | 9.61e-03 |
| 0.6 | 2.054433 | 1.751114 | 3.03e-01 | 2.046176 | 8.26e-03 | 2.035781 | 1.87e-02 |
| 0.7 | 2.664456 | 2.171381 | 4.93e-01 | 2.646934 | 1.75e-02 | 2.6286 | 3.59e-02 |
| 0.8 | 3.59664 | 2.779368 | 8.17e-01 | 3.559596 | 3.70e-02 | 3.527582 | 6.91e-02 |
| 0.9 | 5.05309 | 3.668765 | 1.38e+00 | 4.974892 | 7.82e-02 | 4.91886 | 1.34e-01 |
| 1 | 7.389056 | 4.989521 | 2.40e+00 | 7.223543 | 1.66e-01 | 7.124477 | 2.65e-01 |

**Шаг 0.05**

|  | | **Метод 1** | | **Метод 2** | | **Метод 3** | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **t** | **y\_аналит** | **y\_числен** | **погрешн** | **y\_числен** | **погрешн** | **y\_числен** | **погрешн** |
| 0.0 | 1.0 | 1.0 | 0.00e+00 | 1.0 | 0.00e+00 | 1.0 | 0.00e+00 |
| 0.05 | 1.005013 | 1.0 | 5.01e-03 | 1.005 | 1.25e-05 | 1.005 | 1.25e-05 |
| 0.1 | 1.020201 | 1.01 | 1.02e-02 | 1.020175 | 2.58e-05 | 1.02015 | 5.10e-05 |
| 0.15 | 1.046028 | 1.0302 | 1.58e-02 | 1.045986 | 4.19e-05 | 1.045909 | 1.19e-04 |
| 0.2 | 1.083287 | 1.061106 | 2.22e-02 | 1.083223 | 6.40e-05 | 1.083065 | 2.22e-04 |
| 0.25 | 1.133148 | 1.10355 | 2.96e-02 | 1.133051 | 9.72e-05 | 1.132778 | 3.71e-04 |
| 0.3 | 1.197217 | 1.158728 | 3.85e-02 | 1.197069 | 1.49e-04 | 1.196638 | 5.79e-04 |
| 0.35 | 1.277621 | 1.228251 | 4.94e-02 | 1.277392 | 2.29e-04 | 1.276753 | 8.68e-04 |
| 0.4 | 1.377128 | 1.314229 | 6.29e-02 | 1.376773 | 3.55e-04 | 1.375861 | 1.27e-03 |
| 0.45 | 1.499303 | 1.419367 | 7.99e-02 | 1.498755 | 5.47e-04 | 1.497487 | 1.82e-03 |
| 0.5 | 1.648721 | 1.54711 | 1.02e-01 | 1.647881 | 8.40e-04 | 1.64615 | 2.57e-03 |
| 0.55 | 1.831252 | 1.701821 | 1.29e-01 | 1.829972 | 1.28e-03 | 1.827638 | 3.61e-03 |
| 0.6 | 2.054433 | 1.889022 | 1.65e-01 | 2.052497 | 1.94e-03 | 2.049376 | 5.06e-03 |
| 0.65 | 2.327978 | 2.115704 | 2.12e-01 | 2.325068 | 2.91e-03 | 2.320919 | 7.06e-03 |
| 0.7 | 2.664456 | 2.390746 | 2.74e-01 | 2.660111 | 4.35e-03 | 2.654609 | 9.85e-03 |
| 0.75 | 3.080217 | 2.72545 | 3.55e-01 | 3.073758 | 6.46e-03 | 3.066471 | 1.37e-02 |
| 0.8 | 3.59664 | 3.134268 | 4.62e-01 | 3.587076 | 9.56e-03 | 3.577422 | 1.92e-02 |
| 0.85 | 4.241852 | 3.635751 | 6.06e-01 | 4.227727 | 1.41e-02 | 4.214919 | 2.69e-02 |
| 0.9 | 5.05309 | 4.253829 | 7.99e-01 | 5.032264 | 2.08e-02 | 5.015227 | 3.79e-02 |
| 0.95 | 6.079971 | 5.019518 | 1.06e+00 | 6.049284 | 3.07e-02 | 6.026547 | 5.34e-02 |
| 1 | 7.389056 | 5.973226 | 1.42e+00 | 7.343831 | 4.52e-02 | 7.313366 | 7.57e-02 |

**Шаг 0.025**

|  | | **Метод 1** | | **Метод 2** | | **Метод 3** | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **t** | **y\_аналит** | **y\_числен** | **погрешн** | **y\_числен** | **погрешн** | **y\_числен** | **погрешн** |
| 0.0 | 1.0 | 1.0 | 0.00e+00 | 1.0 | 0.00e+00 | 1.0 | 0.00e+00 |
| 0.025 | 1.001251 | 1.0 | 1.25e-03 | 1.00125 | 7.82e-07 | 1.00125 | 7.82e-07 |
| 0.05 | 1.005013 | 1.0025 | 2.51e-03 | 1.005011 | 1.58e-06 | 1.005009 | 3.14e-06 |
| 0.075 | 1.011314 | 1.007513 | 3.80e-03 | 1.011311 | 2.41e-06 | 1.011306 | 7.13e-06 |
| 0.1 | 1.020201 | 1.015069 | 5.13e-03 | 1.020198 | 3.34e-06 | 1.020189 | 1.28e-05 |
| 0.125 | 1.031743 | 1.02522 | 6.52e-03 | 1.031739 | 4.41e-06 | 1.031723 | 2.04e-05 |
| 0.15 | 1.046028 | 1.038035 | 7.99e-03 | 1.046022 | 5.73e-06 | 1.045998 | 3.00e-05 |
| 0.175 | 1.063165 | 1.053605 | 9.56e-03 | 1.063157 | 7.39e-06 | 1.063123 | 4.19e-05 |
| 0.2 | 1.083287 | 1.072043 | 1.12e-02 | 1.083278 | 9.54e-06 | 1.083231 | 5.63e-05 |
| 0.225 | 1.106553 | 1.093484 | 1.31e-02 | 1.106541 | 1.23e-05 | 1.10648 | 7.36e-05 |
| 0.25 | 1.133148 | 1.118088 | 1.51e-02 | 1.133132 | 1.60e-05 | 1.133054 | 9.43e-05 |
| 0.275 | 1.163287 | 1.14604 | 1.72e-02 | 1.163267 | 2.07e-05 | 1.163169 | 1.19e-04 |
| 0.3 | 1.197217 | 1.177556 | 1.97e-02 | 1.19719 | 2.69e-05 | 1.19707 | 1.48e-04 |
| 0.325 | 1.235221 | 1.212883 | 2.23e-02 | 1.235186 | 3.48e-05 | 1.235039 | 1.82e-04 |
| 0.35 | 1.277621 | 1.252301 | 2.53e-02 | 1.277576 | 4.49e-05 | 1.277399 | 2.22e-04 |
| 0.375 | 1.324785 | 1.296132 | 2.87e-02 | 1.324727 | 5.78e-05 | 1.324515 | 2.70e-04 |
| 0.4 | 1.377128 | 1.344737 | 3.24e-02 | 1.377054 | 7.41e-05 | 1.376802 | 3.25e-04 |
| 0.425 | 1.435122 | 1.398526 | 3.66e-02 | 1.435028 | 9.46e-05 | 1.434731 | 3.91e-04 |
| 0.45 | 1.499303 | 1.457964 | 4.13e-02 | 1.499182 | 1.20e-04 | 1.498835 | 4.68e-04 |
| 0.475 | 1.570274 | 1.523572 | 4.67e-02 | 1.570122 | 1.52e-04 | 1.569715 | 5.58e-04 |
| 0.5 | 1.648721 | 1.595942 | 5.28e-02 | 1.64853 | 1.92e-04 | 1.648057 | 6.65e-04 |
| 0.525 | 1.735421 | 1.675739 | 5.97e-02 | 1.735181 | 2.40e-04 | 1.734631 | 7.90e-04 |
| 0.55 | 1.831252 | 1.763715 | 6.75e-02 | 1.830952 | 3.01e-04 | 1.830315 | 9.37e-04 |
| 0.575 | 1.937212 | 1.860719 | 7.65e-02 | 1.936838 | 3.74e-04 | 1.936101 | 1.11e-03 |
| 0.6 | 2.054433 | 1.967711 | 8.67e-02 | 2.053968 | 4.65e-04 | 2.053118 | 1.32e-03 |
| 0.625 | 2.184201 | 2.085773 | 9.84e-02 | 2.183625 | 5.76e-04 | 2.182644 | 1.56e-03 |
| 0.65 | 2.327978 | 2.216134 | 1.12e-01 | 2.327266 | 7.11e-04 | 2.326135 | 1.84e-03 |
| 0.675 | 2.48743 | 2.360183 | 1.27e-01 | 2.486553 | 8.77e-04 | 2.48525 | 2.18e-03 |
| 0.7 | 2.664456 | 2.519495 | 1.45e-01 | 2.663378 | 1.08e-03 | 2.661878 | 2.58e-03 |
| 0.725 | 2.861225 | 2.69586 | 1.65e-01 | 2.859902 | 1.32e-03 | 2.858175 | 3.05e-03 |
| 0.75 | 3.080217 | 2.89131 | 1.89e-01 | 3.078595 | 1.62e-03 | 3.076606 | 3.61e-03 |
| 0.775 | 3.32427 | 3.108158 | 2.16e-01 | 3.322286 | 1.98e-03 | 3.319995 | 4.28e-03 |
| 0.8 | 3.59664 | 3.34904 | 2.48e-01 | 3.594215 | 2.43e-03 | 3.591575 | 5.06e-03 |
| 0.825 | 3.901067 | 3.616963 | 2.84e-01 | 3.898105 | 2.96e-03 | 3.895063 | 6.00e-03 |
| 0.85 | 4.241852 | 3.915363 | 3.26e-01 | 4.23824 | 3.61e-03 | 4.234731 | 7.12e-03 |
| 0.875 | 4.623953 | 4.248169 | 3.76e-01 | 4.619549 | 4.40e-03 | 4.6155 | 8.45e-03 |
| 0.9 | 5.05309 | 4.619884 | 4.33e-01 | 5.047723 | 5.37e-03 | 5.043046 | 1.00e-02 |
| 0.925 | 5.535877 | 5.035673 | 5.00e-01 | 5.529339 | 6.54e-03 | 5.523932 | 1.19e-02 |
| 0.95 | 6.079971 | 5.501473 | 5.78e-01 | 6.072009 | 7.96e-03 | 6.065752 | 1.42e-02 |
| 0.975 | 6.694257 | 6.024113 | 6.70e-01 | 6.684561 | 9.70e-03 | 6.677313 | 1.69e-02 |
| 1 | 7.389056 | 6.611464 | 7.78e-01 | 7.377249 | 1.18e-02 | 7.368843 | 2.02e-02 |

Порядок аппроксимации - это порядок отношения **погрешн/y\_аналит**.

На всем отрезке первый метод имеет порядок аппроксимации 1.

Второй и третий методы имеют порядок аппроксимации 2.

Уменьшая шага в 2 раза получаем убывающую в 2 раза погрешность, т.е. точность решения растет в 2 раза.

**Код программы**

**import numpy as np**

**import pandas as pd**

**h1, h2, h3 = [0.1, 0.05, 0.025] *# step sizes***

**y = 1.0 *# initial condition***

***# creating grids***

**grid1 = np.arange(0, 1, h1)**

**grid2 = np.arange(0, 1, h2)**

**grid3 = np.arange(0, 1, h3)**

***# defining functions***

**def f(t, y):**

**return 4\*t\*y**

**def f\_solve(t):**

**return np.exp(2\*t\*t)**

## 

## Method 1, h = 0.1

**In [117]:**

**num\_result, real\_result = [], []**

**num\_result.append(y)**

**for i in grid1:**

**y\_new = y + h1\*f(i,y)**

**num\_result.append(y\_new)**

**y\_new\_real = f\_solve(i)**

**real\_result.append(y\_new\_real)**

**y = y\_new**

**y\_new\_real = f\_solve(1)**

**real\_result.append(y\_new\_real)**

**error = np.absolute(np.array(num\_result) - np.array(real\_result))**

**error\_scientific = []**

**for x in error:**

**error\_scientific.append('{:.2e}'.format(x))**

**df1 = pd.DataFrame([grid1, real\_result, num\_result, error\_scientific]).T**

**df1.columns = ['t', 'y\_real', 'y\_num (M1)', '|y\_num-y\_real| (M1)']**

## 

## Method 2, h = 0.1

**In [118]:**

**num\_result = []**

**y = 1.0**

**num\_result.append(y)**

**for i in grid1:**

**y\_new = y + (h1/2.0)\*(f(i,y)+f(i + h1, y + h1\*f(i,y)))**

**num\_result.append(y\_new)**

**y = y\_new**

**error = np.absolute(np.array(num\_result) - np.array(real\_result))**

**error\_scientific = []**

**for x in error:**

**error\_scientific.append('{:.2e}'.format(x))**

**df2 = pd.DataFrame([num\_result, error\_scientific]).T**

**df2.columns = ['y\_num (M2)', '|y\_num-y\_real| (M2)']**

## 

## Method 3, h = 0.1

**In [119]:**

**num\_result = []**

**y = 1.0**

**num\_result.append(y)**

**for i in grid1:**

**y\_new = y + h1\*f(i+(h1/2.0), y + (h1/2.0)\*f(i,y))**

**num\_result.append(y\_new)**

**y = y\_new**

**error = np.absolute(np.array(num\_result) - np.array(real\_result))**

**error\_scientific = []**

**for x in error:**

**error\_scientific.append('{:.2e}'.format(x))**

**df3 = pd.DataFrame([num\_result, error\_scientific]).T**

**df3.columns = ['y\_num (M3)', '|y\_num-y\_real| (M3)']**

## 

## Method 1, h = 0.05

**In [120]:**

**num\_result, real\_result = [], []**

**y = 1.0**

**num\_result.append(y)**

**for i in grid2:**

**y\_new = y + h2\*f(i,y)**

**num\_result.append(y\_new)**

**y\_new\_real = f\_solve(i)**

**real\_result.append(y\_new\_real)**

**y = y\_new**

**y\_new\_real = f\_solve(1)**

**real\_result.append(y\_new\_real)**

**error = np.absolute(np.array(num\_result) - np.array(real\_result))**

**error\_scientific = []**

**for x in error:**

**error\_scientific.append('{:.2e}'.format(x))**

**df4 = pd.DataFrame([grid2, real\_result, num\_result, error\_scientific]).T**

**df4.columns = ['t', 'y\_real', 'y\_num (M1)', '|y\_num-y\_real| (M1)']**

## 

## Method 2, h = 0.05

**In [121]:**

**num\_result = []**

**y = 1.0**

**num\_result.append(y)**

**for i in grid2:**

**y\_new = y + (h2/2.0)\*(f(i,y)+f(i + h2, y + h2\*f(i,y)))**

**num\_result.append(y\_new)**

**y = y\_new**

**error = np.absolute(np.array(num\_result) - np.array(real\_result))**

**error\_scientific = []**

**for x in error:**

**error\_scientific.append('{:.2e}'.format(x))**

**df5 = pd.DataFrame([num\_result, error\_scientific]).T**

**df5.columns = ['y\_num (M2)', '|y\_num-y\_real| (M2)']**

## 

## Method 3, h = 0.05

**In [122]:**

**num\_result = []**

**y = 1.0**

**num\_result.append(y)**

**for i in grid2:**

**y\_new = y + h2\*f(i+(h2/2.0), y + (h2/2.0)\*f(i,y))**

**num\_result.append(y\_new)**

**y = y\_new**

**error = np.absolute(np.array(num\_result) - np.array(real\_result))**

**error\_scientific = []**

**for x in error:**

**error\_scientific.append('{:.2e}'.format(x))**

**df6 = pd.DataFrame([num\_result, error\_scientific]).T**

**df6.columns = ['y\_num (M3)', '|y\_num-y\_real| (M3)']**

## 

## Method 1, h = 0.025

**In [123]:**

**num\_result, real\_result = [], []**

**y = 1.0**

**num\_result.append(y)**

**for i in grid3:**

**y\_new = y + h3\*f(i,y)**

**num\_result.append(y\_new)**

**y\_new\_real = f\_solve(i)**

**real\_result.append(y\_new\_real)**

**y = y\_new**

**y\_new\_real = f\_solve(1)**

**real\_result.append(y\_new\_real)**

**error = np.absolute(np.array(num\_result) - np.array(real\_result))**

**error\_scientific = []**

**for x in error:**

**error\_scientific.append('{:.2e}'.format(x))**

**df7 = pd.DataFrame([grid3, real\_result, num\_result, error\_scientific]).T**

**df7.columns = ['t', 'y\_real', 'y\_num (M1)', '|y\_num-y\_real| (M1)']**

## 

## Method 2, h = 0.025

**In [124]:**

**num\_result = []**

**y = 1.0**

**num\_result.append(y)**

**for i in grid3:**

**y\_new = y + (h3/2.0)\*(f(i,y)+f(i + h3, y + h3\*f(i,y)))**

**num\_result.append(y\_new)**

**y = y\_new**

**error = np.absolute(np.array(num\_result) - np.array(real\_result))**

**error\_scientific = []**

**for x in error:**

**error\_scientific.append('{:.2e}'.format(x))**

**df8 = pd.DataFrame([num\_result, error\_scientific]).T**

**df8.columns = ['y\_num (M2)', '|y\_num-y\_real| (M2)']**

## 

## Method 3, h = 0.025

**In [125]:**

**num\_result = []**

**y = 1.0**

**num\_result.append(y)**

**for i in grid3:**

**y\_new = y + h3\*f(i+(h3/2.0), y + (h3/2.0)\*f(i,y))**

**num\_result.append(y\_new)**

**y = y\_new**

**error = np.absolute(np.array(num\_result) - np.array(real\_result))**

**error\_scientific = []**

**for x in error:**

**error\_scientific.append('{:.2e}'.format(x))**

**df9 = pd.DataFrame([num\_result, error\_scientific]).T**

**df9.columns = ['y\_num (M3)', '|y\_num-y\_real| (M3)']**

## 

**In [126]:**

**df1.loc[10, 't'], df4.loc[20, 't'], df7.loc[40, 't'] = 1, 1, 1**

**h1\_results = pd.concat([df1, df2, df3], axis=1)**

**h2\_results = pd.concat([df4, df5, df6], axis=1)**

**h3\_results = pd.concat([df7, df8, df9], axis=1)**